

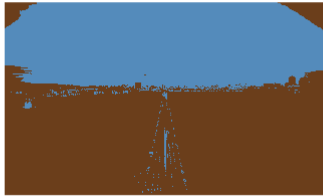
Image Understanding Homework #3

Kishore Narendran – 14644574 – narendrk@uci.edu

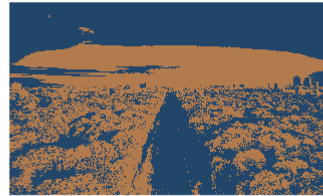
Color Quantization

The reduction of color palette using k means clustering on images, is as described below. The function is performed in the **homework3_5.m** and the **color_quantization.m** file. The result of performing this k means clustering based on RGB values, with values of k as 2,5, 10 is as described below. The result of scaling red is as shown by the images on the right column.

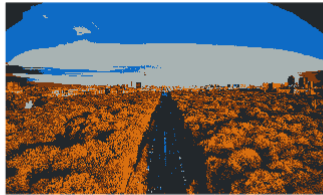
k Means Clusering with k = 2



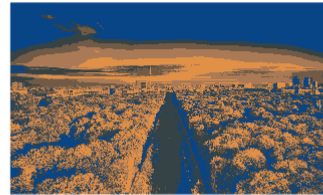
k Means Clusering of RGB with k = 2 (Scaled Red)



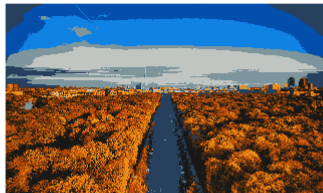
k Means Clusering with k = 5



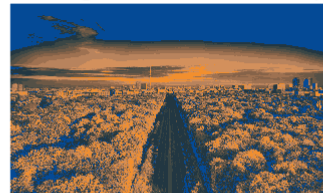
k Means Clusering of RGB with k = 5 (Scaled Red)



k Means Clusering with k = 10



k Means Clusering of RGB with k = 10 (Scaled Red)



The clustering was performed on the following image.



When scaling the red channel, it effectively pushes the pixels with higher red content into another cluster by pushing them away from the regular pixel intensity distributions. What follows, is that, objects in the image, with some red components always get clustered together earlier with even lower values of k . And, pixels with no red content tend to be clustered away from ones that have red.

Filterbank

The code for forming the filterbank and to get the responses of an image using the filter bank is as shown in the following files.

homework3_2.m

```
img = im2double(rgb2gray(imread('img1.jpg')));

% Finding the filtering of image using the derivative of a Gaussian
% in the x and y direction for different values of sigma
% Also returns the 2D Gaussian Filter
gauss_filt1 = derivative_of_gaussian(img, 1);
gauss_filt2 = derivative_of_gaussian(img, 2);
gauss_filt4 = derivative_of_gaussian(img, 4);

% Forming Center Surround Filters as a difference of two gaussian functions
center_surround_filter1 = gauss_filt2 - gauss_filt1;
center_surround_filter2 = gauss_filt4 - gauss_filt2;

% Displaying the result of using the center surround filter
img_cs1 = imfilter(img, center_surround_filter1);
img_cs2 = imfilter(img, center_surround_filter2);

figure;
subplot(1,2,1);
imagesc(center_surround_filter1);
colorbar;
colormap jet;
title('Center Surround Filter - 1');
subplot(1,2,2);
imagesc(center_surround_filter2);
colorbar;
colormap jet;
title('Center Surround Filter - 2');
print('output/output_3_2_csf_kernel', '-dpng');
close;

figure;
subplot(1,3,1);
imshow(img);
title('Grayscale Image');
subplot(1,3,2);
imshow(mat2gray(img_cs1));
title('Center Surround Filtered - 1');
subplot(1,3,3);
imshow(mat2gray(img_cs2));
title('Center Surround Filtered - 1');
```

```

print('output/output_3_2_csf', '-dpng');
close;
derivative_of_gaussian.m

function [ gauss_filt ] = derivative_of_gaussian( img, sigma )
    gauss_filt = fspecial('gaussian', [ 5 5 ], sigma);
    [ gauss_filtx, gauss_filty ] = gradient(gauss_filt);

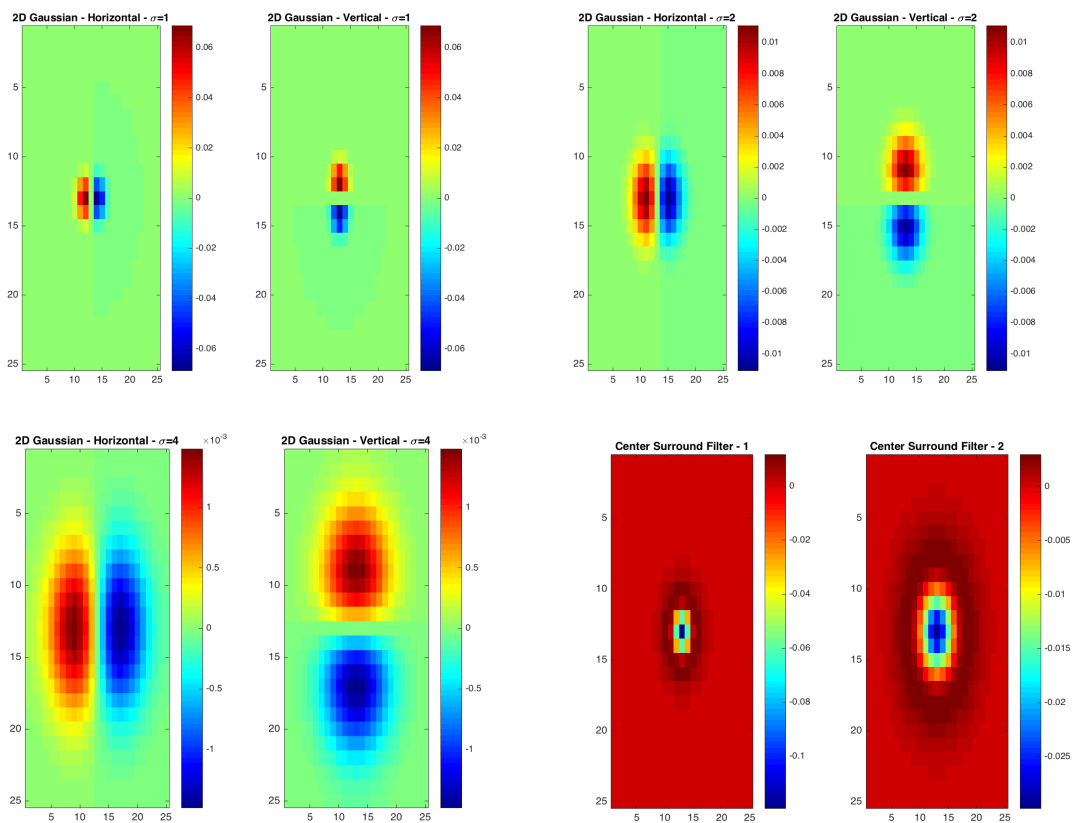
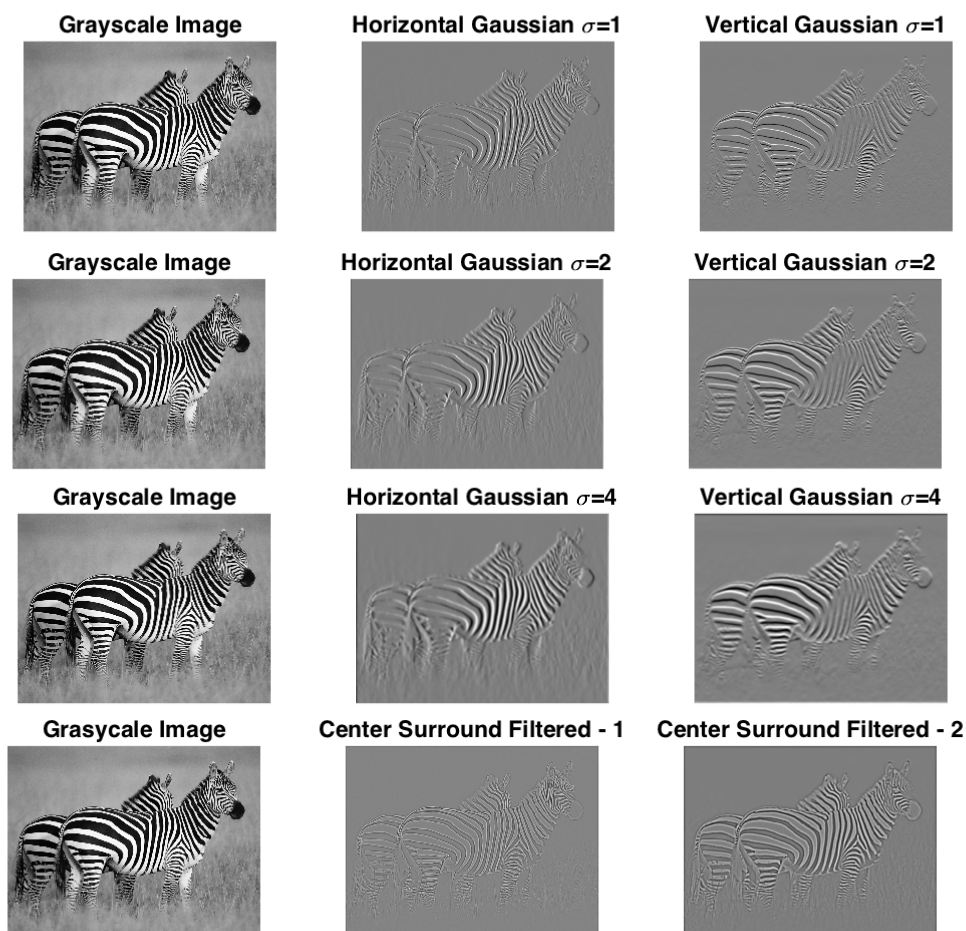
    figure;
    subplot(1,2,1);
    imagesc(gauss_filtx);
    title(strcat('2D Gaussian - Horizontal - \sigma=', int2str(sigma)));
    colorbar;
    colormap jet;
    subplot(1,2,2);
    imagesc(gauss_filty);
    title(strcat('2D Gaussian - Vertical - \sigma=', int2str(sigma)));
    colorbar;
    colormap jet;
    print(strcat('output/output3_2_', 'filter_kernel_', int2str(sigma)), '-dpng');
    close;

    img_gfx = imfilter(img, gauss_filtx);
    img_gfy = imfilter(img, gauss_filty);

    figure;
    subplot(1,3,1);
    imshow(img);
    title('Grayscale Image');
    subplot(1,3,2);
    imshow(img_gfx);
    title(strcat('Horizontal Gaussian \sigma=', int2str(sigma)));
    subplot(1,3,3);
    imshow(img_gfy);
    title(strcat('Vertical Gaussian \sigma=', int2str(sigma)));
    print(strcat('output/output3_2_', 'gauss_sig', int2str(sigma)), '-dpng');
    close;
end

```

The result of filtering the image using the filterbank and the visualization of the kernels is as shown on the following page.

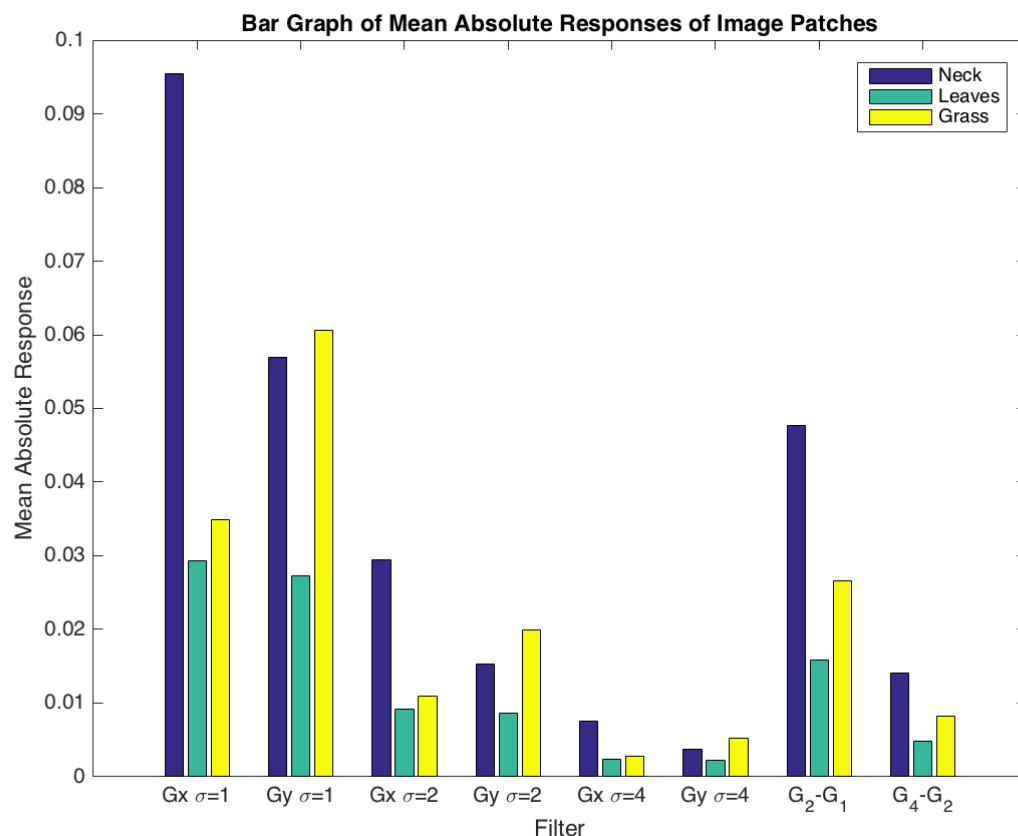


Filter Distributions

The mean absolute responses of the filterbank was found for the following three 40x40 patches in the Zebra image. This is the neck, leaves and grass image respectively.



The bar graph of the mean absolute responses for the image patches when performing filtering using the filterbank is as shown below.

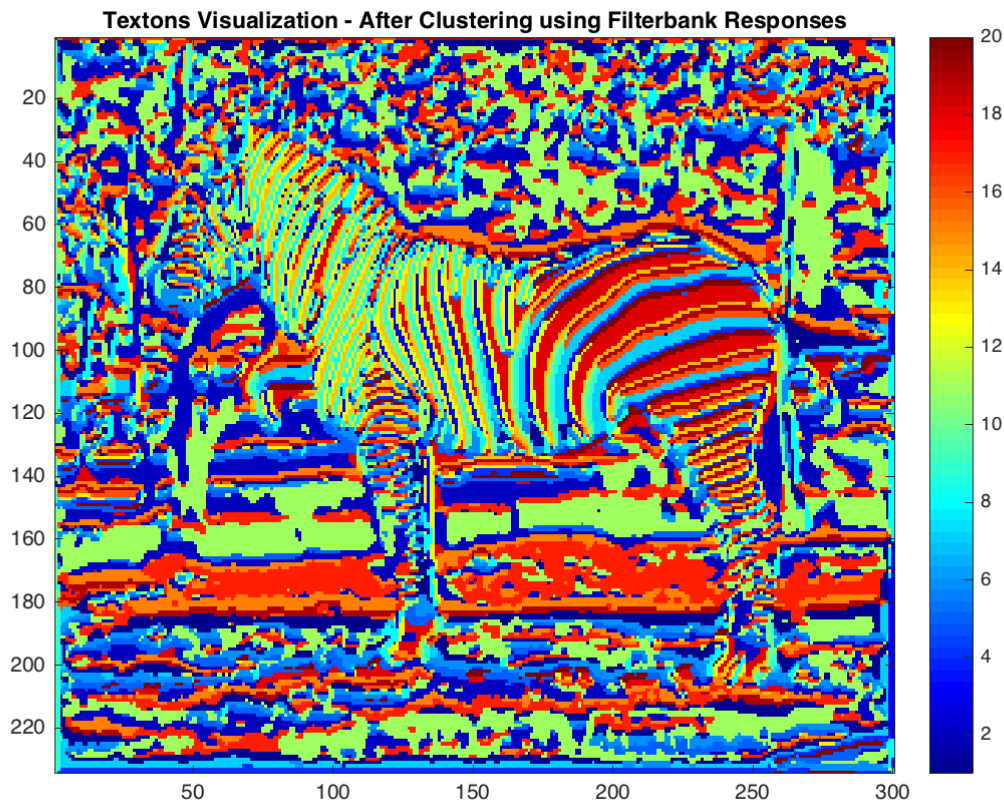


The neck image has a lot of vertical stripes and hence has high values for the mean absolute response for the x directional Gaussian filter along various sigma values. The leaves have no significant edges that are visible along either the x or the y direction and hence show no high

responses with any of the filters in the filterbank. The grass seems to have some edges along the horizontal axes and hence shows some higher value than the zebra's neck along the Gaussian derivative in the y direction.

Textons

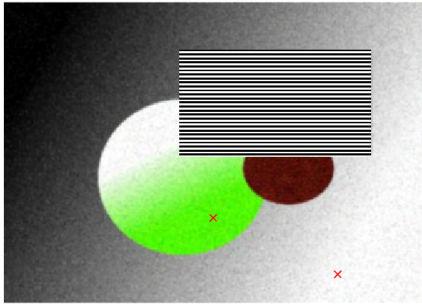
The texton visualization for $k=20$ is as shown below.



MRF Visualization

The foreground and background separation when done using the difference in the RGB values achieved the following with images segtest1, segtest2 and the image used in the filterbank question 2. The results are as shown below.

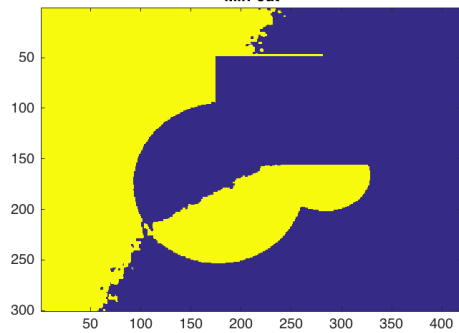
Original image



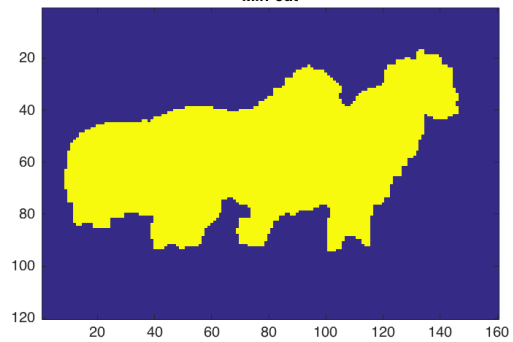
Original image



Min-cut



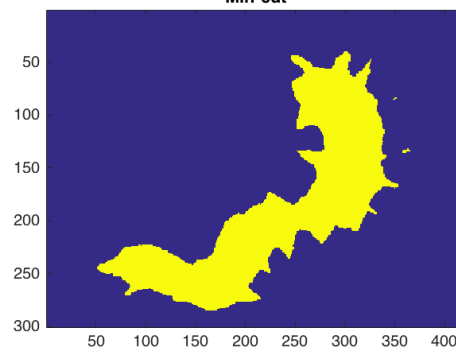
Min-cut



Original image



Min-cut



This approach of differentiating the foreground and the background, seems to work well when there is one object in the foreground which has a certain consistency of colour. However, when there are multiple objects with different pixel colour distributions then this approach doesn't seem to work as well.